Dimensionality reduction

e.g. images ⟶ input space $X \subseteq \mathbb{R}^{w \times h \times d}$ ⟶ in principle we have such many degrees of freedom. This means that any value $\in \mathbb{R}^{w \times h \times d}$ could potentially be a valid element of the dataset. This is not true. There are often not so many degrees of freedom in the variability of the data in the dataset e.g. not all $w \times h \times d$ images contain real information, some are only noise, a bunch of random pixels.

We should make a distinction between the dimension of the input space and the actual variability of the data in the dataset.

We want to study how to build and use a representation of the data in a much smaller dimensional space; the new representation should not lose important information. $\mathbb{R}^{w \times h \times d} \longrightarrow \mathbb{R}^{d \ll w \times h \times d}$.

Dimensionality reduction: understand how to transform a problem in many dimensions to a problem in much less dimensions with the goal of keeping as much as possible the information. Sometimes information loss is inevitable.

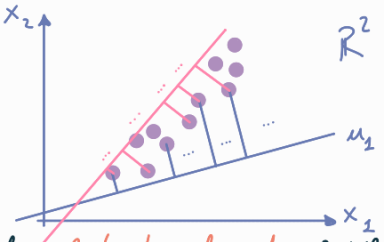
Principal Component Analysis (PCA)

PCA can only discover linear transformations

PCA is a dimensionality reduction method.

Given data $\{x_n\} \in \mathbb{R}^N$, the goal is to **maximize variance** after projection to some direction u. $\mathbb{R}^N \longrightarrow \mathbb{R}^D$ with $D < N$. Output vector will be $< u_1^T x_n, \dots, u_i^T x_n, \dots, u_d^T x_n >$ with $< u_1, \dots, u_d >$ are basis vectors for $\mathbb{R}^D$ ($u_i^T u_j = 1 \ \forall i \neq j$ ⟶ vectors are orthogonal)

x_1 and x_2 are the variables of the input space

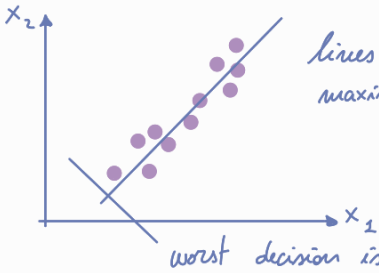


$\mathbb{R}^2 \longrightarrow \mathbb{R}^1$

output space has fewer dimensions

The point is: what is the **best transformation**? What transformation **minimizes the error** (maximum information retain)?

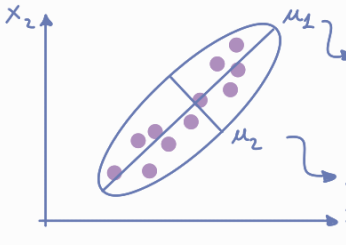
If the data is somewhat aligned (like in the previous example) the choice of the direction is easy



lines parallel to this guarantee maximum information retain

worst decision is to use a line perpendicular to the alignment of the data points

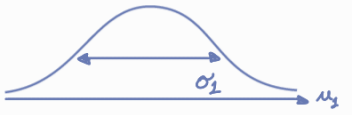
The general idea is to compute the covariance of the dataset and choose the direction of the maximum variance.



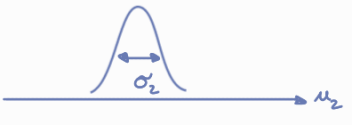
u_1 → maximum variance: the data is maximally separated

u_2 → minimal variance

projection on u_1:



projection on u_2:



if you consider any other direction you will have an intermediate solution $\sigma_2 < \sigma < \sigma_1$

find the base vectors that yields the best transformation

$\mathbb{R}^M \xrightarrow{\text{projection}} \mathbb{R}^D$

$\langle x_1, \dots, x_M \rangle \quad \langle u_1, \dots, u_D \rangle$

we need to project from M dimensions to D dimensions

$\langle u_1^T x_m, u_2^T x_m, \dots, u_D x_m^T \rangle \in \mathbb{R}^D$

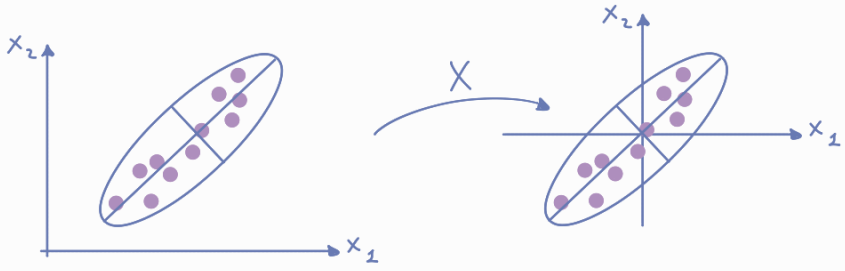
$\langle u_1, \dots, u_D \rangle$ is a base $\longrightarrow u_i^T u_j = 1 \quad \forall i, j, i \neq j$

We just need to find $\langle u_1, \dots, u_D \rangle$, a particular set of base vectors that maximize the variance of the projected points. In order to do this we define:

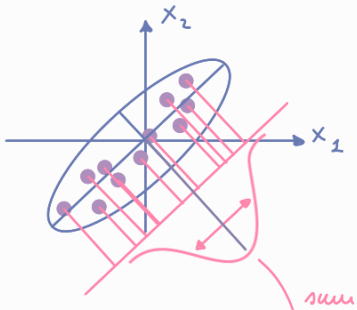
mean value of the data points $\quad \bar{x} = \frac{1}{N} \sum_{n=1}^{N} x_n$

data-centered matrix $\quad X = \begin{bmatrix} (x_1 - \bar{x})^T \\ \dots \\ (x_i - \bar{x})^T \\ \dots \\ (x_N - \bar{x})^T \end{bmatrix}$

matrix with which we normalize the data so that the mean of the dataset matrix will be zero. This is equal to changing the reference system so that the mean is the new center.



Now compute the variance of the projected points; the new mean is the origin (thanks to the data centered matrix)



mean of projected points $\quad u_1^T \bar{x}$

variance of projected points $\quad \frac{1}{N} \sum_{n=1}^{N} (u_1^T x_n - u_1^T \bar{x})^2 = u_1^T S u_1$

sum of squared differences

only one dimension

data centered matrix

with $\quad S = \frac{1}{N} \sum_{n=1}^{N} (x_n - \bar{x})(x_n - \bar{x})^T = \frac{1}{N} X^T X$

matrix S contains the variance of all the points in the dataset

Problem definition becomes: $\max_{u_1} u_1^T S u_1$ subject to constraint $u_1^T u_1 = 1$

e.g. solve with Lagrange multipliers

By solving this optimization problem we come to this solution:

$u_1^T S u_1 = \lambda_1$

when the variance of the projected points is maximum, we have this relation:

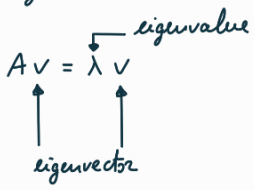
λ_1 is an eigenvalue of S, u_1 is an eigenvector of S $\longrightarrow$ $S u_1 = \lambda_1 u_1$

maximum value (solution) corresponds to the eigenvalue of S, so we need to find the highest eigenvalue.

the maximum value $u_1^T S u_1$ is exactly λ_1.
If we had a bidimensional space ($\mathbb{R}^N \rightarrow \mathbb{R}^2$), S will have two eigenvalues λ_1 and λ_2 and the solution of the maximum variance problem is the one that corresponds to the highest eigenvalue.
This can be generalized in N dimensions.
If we order the eigenvalues in decreasing order, the first one will maximize the variance in the first dimension, the second one will maximize the variance in the second dimension and so on ...

Eigenvalues and Eigenvectors

For a square matrix A, an eigenvector and eigenvalue make this equation true:



$$\begin{bmatrix} -6 & 3 \\ 4 & 5 \end{bmatrix} \qquad \begin{bmatrix} 1 \\ 4 \end{bmatrix} \qquad 6$$

matrix eigenvector eigenvalue

$$A \cdot v = \begin{bmatrix} -6 & 3 \\ 4 & 5 \end{bmatrix} \begin{bmatrix} 1 \\ 4 \end{bmatrix} = \begin{bmatrix} -6 \cdot 1 & 3 \cdot 4 \\ 4 \cdot 1 & 5 \cdot 4 \end{bmatrix} = \begin{bmatrix} 6 \\ 24 \end{bmatrix}$$

$$\lambda v = 6 \begin{bmatrix} 1 \\ 4 \end{bmatrix} = \begin{bmatrix} 6 \\ 24 \end{bmatrix}$$

Solution

$$u_1^T S u_1 = \lambda_1$$

Variance is maximal when u_1 is the eigenvector corresponding to the largest eigenvalue λ_1. This is called *principal component*. The direction u_1 is the corresponding eigenvector.

Approach:

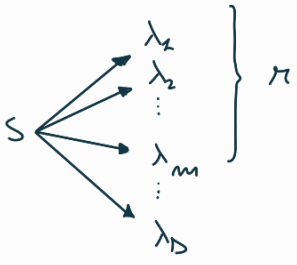
We start from the dataset of dimension D; we want a dataset with dimension $M < D$.

① From the dataset we generate the data centered matrix

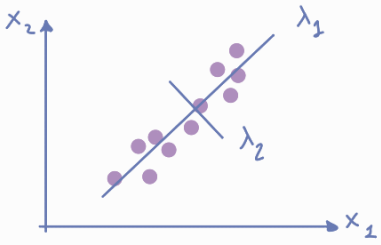
② From the new data we generate the mean and variance ⟶ $u_1^T S u_1$ is the variance

③ We maximize the variance $\max_{u_1} u_1^T S u_1$

Solution to this maximization problem can be found computing the eigenvalues of S, ordering them from the highest to the lowest and taking the first M out of them.



This is still a *linear transformation*.



$\lambda_1 > \lambda_2$

λ_1 } eigenvalues

λ_2

$D=2$ -dimensional input space

In general, when we have d dimensions with $d > 2$ we will have d eigenvalues. The highest eigenvalues correspond to the direction in which the projected points have maximum variance.

Once we project all the data on the first eigenvector we reduce the dataset from dimension d to dimension $d-1$; we can iterate this procedure. If you do this and repeat this procedure a second time you will obtain the second best eigenvector. In general, what we can do in order to reduce the space from D to M ($D > M$) is to:

① compute $\bar{x}$: mean of the data;
② compute S: covariance matrix of the data;
③ find M eigenvectors of S corresponding to the M largest eigenvalues.

Consider another formulation of the problem (transformation from a D-dimensional space to another D-dimensional space)

Given a d-dimensional dataset and a complete orthonormal d-dimensional basis such that

dot product $u_i \cdot u_j = u_i^T u_j = |u_i||u_j| \cos\theta$

is $= 0$ when $\cos\theta = 0 \longleftrightarrow \theta = 90$

$\longleftrightarrow$ u_i and u_j are perpendicular

$\longleftrightarrow$ linearly independent

$\longleftrightarrow$ part of a base for the space

$$u_i^T u_j = \delta_{ij}$$

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

d-dimensional space + d-dimensional base: we can consider the u vectors as a new reference space in which we want to transform the original dataset (new base)

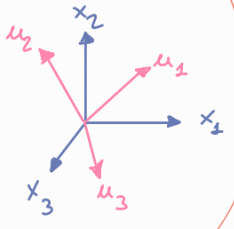
each datapoint can be written as

$$x_n = \sum_{i=1}^{D} \alpha_{ni} u_i$$

representation of x in terms of the new base $(u_1, \dots, u_D)$

Using the orthonormality property we have $\alpha_{nj} = x_n^T u_j$, hence:

$$x_n = \sum_{i=1}^{D} (x_n^T u_i) u_i$$



The new basis has the same dimension of the original space. This is only a transformation

Goal: approximate x_n using a lower dimensional representation

Split the above definition in two parts:

$$\tilde{x}_n = \sum_{i=1}^{M} z_{ni} u_i + \sum_{i=M+1}^{D} b_i u_i$$

first part is derived from the actual value in the dataset

second part is computed from the mean

$z_{nj} = x_n^T u_j \quad j = 1, \dots, M$ → this part is correct

mean value

$b_j = \bar{x}^T u_j \quad j = M+1, \dots, D$ → this part is an approximation

If we find a good way to approximate them we will only need the first M components.

$\tilde{x}_n$ needs to be as close as possible to x_n.

Evaluate approximation error as:

$$J = \frac{1}{N} \sum_{n=1}^{N} \| x_n - \tilde{x}_n \|^2$$

The error in the approximation is given only by the last $D-M$ components, because the first M components are identical in x and $\tilde{x}$.

$$x - \tilde{x} = \sum_{i=M+1}^{D} [(x_n - \bar{x})^T u_i] u_i$$

the $D-M$ components are derived from the mean

We consider this $(x_m - \tilde{x}_m)$ as an error just for sample x_m

Putting together this procedure for each sample:

$$J = \frac{1}{N} \sum_{m=1}^{N} \sum_{i=M+1}^{D} (x_m^T u_i - \bar{x}^T u_i)^2 = \sum_{i=M+1}^{D} u_i^T S u_i$$

- sum of squared errors for all the samples in the dataset where the error is considered only on the last $D-M$ components
- normalize for all the samples in the dataset
- squared error
- covariance matrix

We have the same maximization problem that we had when we had to maximize the variance of the projection of each point, so we have the same solution:

$$S u_i = \lambda_i u_i$$

but since we have to minimize the error while before we had to maximize the variance, now we choose the lowest $D-M$ eigenvalues.

ordered from high to low

$$S = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_D \end{bmatrix} \begin{matrix} \} M \\ \} D-M \end{matrix}$$

- the first M are the ones that maximize the variance.
- the last $D-M$ are the ones that minimize the approximation error.

We use the first M highest eigenvalues to project the dataset on this M dimensions and we automatically obtain that the result correspond to the minimization of the approximation error.